

REMARKS

The Office action has been carefully considered. The Office action rejected claims 1, 3-9, 11-29, and 31-37 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,819,263 to Bromley et al. ("Bromley"). Applicants respectfully disagree.

Applicants thank the Examiner for the interview held (by telephone) on February 10, 2004. During the interview, the Examiner and applicants' attorney discussed the claims with respect to the prior art. The essence of applicants' position is incorporated in the remarks below.

Please note that in previous Office actions and Office action responses, certain claims were erroneously referred to by the incorrect numbers. In essence, claims 17-19 were referenced instead of claims 18-20 and claims 20-33 were referenced instead of claims 21-33 by both the previous Office actions and previous Office action responses. The arguments and the language used in both the previous Office actions and subsequent responses both referred to the correct language but did not refer to the correct claim numbers. As such, a listing of the claims is presented (though no amendments are made at present) that show the current status of the claims. All remarks below refer to the true and correct claim numbers.

Prior to discussing reasons why applicants believe that the claims in this application are clearly allowable in view of the teachings of the cited and applied references, a brief description of the present invention is presented.

The present invention is generally directed toward a financial or other planning system and method in which hierarchically arranged objects are created and maintained to form a plan. The hierarchical arrangement enables objects to be dependent on other objects, while within the objects are fields that can be related to other fields, *e.g.*, dates, dollar amounts, interest rates and so on. Significantly, the user of the system and method need not be concerned with the dependencies / relationships among objects and fields, but rather simply selects elements and enters data for those elements, and thereafter lets the objects of the system and method handle the dependencies. Thus, unlike simple programming techniques, a user simply responds to questions via a user interface, fills in information and/or makes selections related to the plan. The system then writes the proper information into the hierarchically arranged objects for the user, and manages the relationships for the user. A planning engine runs a simulation based on the data in the objects.

The combination of hierarchical objects relationships and relative field values allows a great deal of flexibility in creating what may be a very complex data system, which can then be used to calculate the results of the user's financial plan over time, as well as making it fairly straightforward for the user to make changes to and update a plan.

Moreover, via simple interaction with the user interface, a user can selectively disable objects and/or fields, which automatically disables those objects and fields that are dependent on the directly disabled ones. This facilitates the

running of various "what-if" type simulations, to determine the expected consequences of various possible actions.

Note that the above description is for informational purposes only, and should not be used to interpret the claims, which are discussed below.

Turning to the claims, independent claim 1 recites a computer-readable medium having computer-executable instructions, comprising, receiving input of a value corresponding to a first field of a first object that maintains plan data, receiving additional input corresponding to a second field of a second object that maintains plan data, receiving input that defines a hierarchical relationship between the first and second objects such that a value in the second field is at least partially based on the first field as a result of the hierarchical relationship, developing a plan by running a simulation on objects that maintain the plan data including the first and second objects, receiving input of a new value for the first field, and developing a new plan by running a simulation on objects that maintain the plan data, including the first and second objects, in which in the new plan, the new value changes the information in the second field.

The Office action rejected claim 1 as unpatentable over Bromley. More specifically, the Office action contends that Bromley teaches receiving input of a value corresponding to a first field of a first object that maintains plan data as recited in claim 1. Column 20, lines 15-25 of Bromley are referenced. However, the Office action admits that Bromley does not disclose *any* of the remaining elements in claim 1. Instead, the Office action contends that the remaining elements are known to those skilled in the art as basic programming standards.

The Office action attached its own generated figures 1-4 as illustrative of basic programming standards. In essence, the Office action concludes that it would have been obvious to a programmer (one skilled in the art) at the time the invention was made to take the teachings of Bromley (*i.e.*, receiving input of a value corresponding to a first field of a first object that maintains plan data) and modify the method of Bromley to include the remaining limitations in claim 1. That is, the Office action contends that the steps of (1) receiving additional input corresponding to a second field of a second object that maintains plan data, (2) receiving input that defines a hierarchical relationship between the first and second objects such that a value in the second field is at least partially based on the first field as a result of the hierarchical relationship, (3) developing a plan by running a simulation on objects that maintain the plan data including the first and second objects, (4) receiving input of a new value for the first field, and (5) developing a new plan by running a simulation on objects that maintain the plan data, including the first and second objects, in which in the new plan, the new value changes the information in the second field, are all steps that would have been obvious to a person skilled in the art at the time of the invention. Applicants respectfully disagree.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The

teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). As discussed in greater detail below, Bromley, taken separately or in conjunction with known programming standards at the time of the invention does not teach, suggest, or provide any motivation to arrive at each of the claim limitations.

Briefly, the cited and applied reference is directed toward a management tool that groups various types of financial clients and prospective clients together for an advisor to use in communicating advice and in marketing products to those clients and prospects. As noted by the Office action, Bromley fails to teach hierarchically arranged objects to represent a plan, receiving input that defines hierarchical relationships between objects, running a simulation on those objects, and/or having changes to an object's state data affect other objects hierarchically below that changed object. Further, Bromley fails to teach selectively disabling objects (or the fields therein) for the purpose of running a simulated plan.

Specifically, the Office action contends that 5 of the 6 limitations of claim 1 are known programming standards. The Office action's contention that these limitations are well known and therefore "could be" programmed, however, fails to consider that having this information in related fields of hierarchically arranged objects *eliminates the need for a user to program such formulas*. In fact, one aspect of the present invention is specifically directed towards eliminating the need for such customized software programming or complex formula management as

suggested by the Office action. The Office action provides no support for developing any plan by running a simulation on objects including the first and second objects much less a new plan based on new information that is input to one of the objects.

Further, there is no showing in the prior art of record (specifically, the Office action's figures) that teaches receiving input that defines a hierarchical relationship between the first and second objects such that a value in the second field is at least partially based on the first field as a result of the hierarchical relationship. At best, the Office action's figures show two fields wherein the value of the second field is mathematically based on the value of the first field. Using the example of the Office action's figures, a user may enter, in the first field, a "year of birth." Then, the value in the second field (the second field having a mathematical relationship, *i.e.*, YOB+65 as opposed to a hierarchical relationship) is automatically calculated based on the input of the first field.

Further, the figures of the Office Action may even be interpreted to having a first input in a first field and a second input in a second field. However, it cannot be supported that a third input (*i.e.*, input that defines a hierarchical relationship between the first and second objects as recited in claim 1) is shown by the Office action figures or is simply a known programming standard. Neither Bromley, nor the Office action figures teach, show, suggest or even have any appreciation of establishing a hierarchical relationship between the two object fields let alone using the input in a plan.

To illustrate further by way of a contrasting example supported by the recitations of claim 1, a system may receive (a first time) in the first field, a "current debt" that is a combination of several accounts. The system may then receive input (a second time) in a second field, "a new home purchase date." Having these two fields established, the system may then receive input (a third time) that sets up a hierarchical relationship between the two fields, such as adjusting the "new home purchase date" forward or backward according to the value of "current debt" if the current debt rises above or falls below threshold levels. As such, the relationship between the fields is not purely mathematical but rather hierarchical in that the second field "new home purchase date" does not necessarily change if "current debt" changes. Rather, it behaves according to the hierarchical relationship that was established. Clearly, the recitations of claim 1 are not taught as known programming standards in the Office action's figures 1-4..

The Office action also contends that the motivation to combine the teachings of Bromley with known programming standards is found because it would provide the system of Bromley the flexibility to create and update the financial plan. This is flawed reasoning. Using an analogy, this reasoning would also suggest that any improvement to a car's fuel efficiency would be rendered obvious because it is desirable to have a more fuel efficient car. The motivation to solve a problem does not render the solution to the problem obvious.

Applicants submit that claim 1 is allowable over the prior art of record for at least these reasons.

With regard to claims 2-9, 11-16, and 34, these claims depend either directly or indirectly from claim 1. Applicants submit that claims 2-9, 11-16, and 34 are also allowable for the additional patentable elements included in these claims.

For example, claims 5 and 6 were rejected, despite the Office Action having explicitly conceded that Bromley does not teach an amount in one field with a date in another field conditional on the amount. The Office Action's contention that this is well known and therefore "could be" programmed, however, fails to consider that having this information in related fields of hierarchically arranged objects eliminates the need for a user to program such formulas. In fact, this aspect of the present invention is specifically directed towards *eliminating the need for such customized software programming or complex formula management* as suggested by the Office action.

Turning to claims 17-19, and 35, these claims were inadvertently referred to as claims 18-20 and 35 in the current and previous Office actions. However, these claims are discussed herein referring to the correct numbering. As stated by the Office action, these claims were rejected for similar reasons as given for claims 1, 12-14, and 16. Applicants respectfully disagree.

Independent claim 17 recites in a computer system, a method of organizing information related to a plan, comprising, providing access to a limited number of objects to a user, each object having fields therein for maintaining plan information, receiving first user input information including a value associated with a first field of a first object, and receiving second user input information associated with a second field of a second object, the second input information specifying a relationship of

the second field with the first field, disabling at least one object, and developing a plan including running a simulation that excludes each disabled object.

As was shown above, the Office action's contention that these limitations are well known and therefore "could be" programmed, is an erroneous conclusion. The Office action fails to consider that having this information in related fields of objects *eliminates the need for a user to program such formulas*. In fact, one aspect of the present invention is specifically directed towards eliminating the need for such customized software programming or complex formula management as suggested by the Office action.

Further, claim 17 recites disabling at least one object, and developing a plan including running a simulation that excludes each disabled object. There is no showing in the prior art of record (specifically, the Office action's figures) that teaches running a simulation, let alone developing a plan including running a simulation that excludes each disabled object. At best, the Office action's figures show two fields wherein the value of the second field is mathematically based on the value of the first field. It simply cannot be supported that the additional limitations of claim 17 are known programming standards. Applicants submit that claims 17-19 and 35 are allowable over the prior art of record for at least these additional reasons.

Turning to claims 20-33 and 37, these claims were inadvertently referred to as claims 21-33 and 37 in the current and previous Office actions. However, these claims are discussed herein using the correct numbering. As stated by the Office

action, these claims were rejected for similar reasons as given for claims 1, 3-9, 12-14, 16, 17 and 34. Applicants respectfully disagree.

Independent claim 20 recites a system for outputting a plan, comprising, a user interface for presenting a limited number of plan objects to a user and for receiving data for a first field of a first plan object and data for a second field of a second plan object, the data of the second field having a value linked to the data of the first field via a hierarchical relationship between the first and second objects, the user interface further providing a mechanism that allows plan objects to be selectively disabled, and a planner engine for developing a plan by running a simulation on plan objects while excluding any disabled plan objects.

As was shown above, the Office action's contention that these limitations are well known and therefore "could be" programmed, is an erroneous conclusion. The Office action fails to consider that having this information in related fields of hierarchically arranged objects *eliminates the need for a user to program such formulas*. In fact, one aspect of the present invention is specifically directed towards eliminating the need for such customized software programming or complex formula management as suggested by the Office action.

Further, claim 20 recites a mechanism that allows plan objects to be selectively disabled, and a planner engine for developing a plan by running a simulation on plan objects while excluding any disabled plan objects. Again, there is no showing in the prior art of record (specifically, the Office action's figures) that teaches running a simulation, let alone developing a plan including running a simulation that excludes each disabled object. At best, the Office action's figures

show two fields wherein the value of the second field is mathematically based on the first field. It cannot be supported that the limitations of claim 20 are known programming standards. Applicants submit that claims 20-33 and 37 are allowable over the prior art of record for at least these additional reasons.

Finally, claim 36 recites a computer-readable medium having computer-executable instructions, comprising, providing access to a limited number of objects to a user, each object having fields therein for maintaining plan information, receiving first user input information including a value associated with a first field of a first object, receiving second user input information associated with a second field of a second object, the second input information specifying a relationship of the second field with the first field, disabling at least one object, and developing a plan including running a simulation that excludes each disabled object.

The Office action rejected claim 36 for similar reasons to that of claim 1. As was shown above, the Office action's contention that these limitations are well known and therefore "could be" programmed, is an erroneous conclusion. The Office action fails to consider that having this information in related fields of objects *eliminates the need for a user to program such formulas*. In fact, one aspect of the present invention is specifically directed towards eliminating the need for such customized software programming or complex formula management as suggested by the Office action.

Further, claim 36 recites developing a plan including running a simulation that excludes each disabled object. Again, there is no showing in the prior art of record (specifically, the Office action's figures) that teaches running a simulation,

let alone developing a plan including running a simulation that excludes each disabled object. At best, the Office action's figures show two fields wherein the value of the second field is mathematically based on the first field. It simply cannot be supported that the limitations of claim 36 are known programming standards. Applicants submit that claim 36 is allowable over the prior art of record for at least these additional reasons.

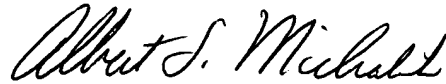
For at least the foregoing reasons, applicants submit that all the claims are patentable over the prior art of record. Reconsideration and withdrawal of the rejections in the Office Action is respectfully requested and early allowance of this application is earnestly solicited.

CONCLUSION

In view of the foregoing remarks, it is respectfully submitted that claims 1, 3-9, 11-29, and 31-37 are patentable over the prior art of record, and that the application is good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,



Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)